Pandora: A Tool to Estimate Dimensionality Reduction Stability of Genotype Data

Julia Haag[2], Alexander I. Jordan[3], and Alexandros Stamatakis[1,2,4]

[1]Biodiversity Computing Group, Institute of Computer Science, Foundation for Research and Technology - Hellas, Heraklion, Crete, Greece
[2]Computational Molecular Evolution Group, Heidelberg Institute for Theoretical Studies, Heidelberg, Germany
[3]Computational Statistics Group, Heidelberg Institute for Theoretical Studies, Heidelberg, Germany
[4]Institute for Theoretical Informatics, Karlsruhe Institute of Technology, Karlsruhe, Germany

August 2024

Abstract

Motivation: Genotype datasets typically contain a large number of single nucleotide polymorphisms for a comparatively small number of individuals. To identify similarities between individuals and to infer an individual's origin or membership to a cultural group, dimensionality reduction techniques are routinely deployed. However, inherent (technical) difficulties such as missing or noisy data need to be accounted for when analyzing a lower dimensional representation of genotype data, and the intrinsic uncertainty of such analyses should be reported in all studies. However, to date, there exists no stability assessment technique for genotype data that can estimate this uncertainty.

Results: Here, we present Pandora, a stability estimation framework for genotype data based on bootstrapping. Pandora computes an overall score to quantify the stability of the entire embedding, infers per-individual support values, and also deploys a k-means clustering approach to assess the uncertainty of assignments to potential cultural groups. In addition to this bootstrap-based stability estimation, Pandora offers a sliding-window stability estimation for whole-genome data. Using published empirical and simulated datasets, we demonstrate the usage and utility of Pandora for studies that rely on dimensionality reduction techniques.

Availability and Implementation: Pandora is available on GitHub `https://github.com/tschuelia/Pandora`.

Contact: `julia.haag@h-its.org`

Supplementary information: All Python scripts and data to reproduce our results are available on GitHub `https://github.com/tschuelia/PandoraPaper`**.**

1 Introduction

Dimensionality reduction techniques such as Principal Component Analysis (PCA) and Multidimensional Scaling (MDS) are widely deployed in numerous scientific fields as they facilitate data visualization and interpretation. Both methods project high dimensional data to lower dimensions, while preserving as much variation, and thus information, as possible. Ever since the first application of PCA to genetic data almost half a century ago [10], PCA as well as alternative dimensionality reduction techniques have been widely used in modern population genomic analyses, including ancient DNA studies, to draw conclusions about population structure, genetic variation, or demographic history. For example, Hughey et al. [8] found that the Minoan civilization did not originate from Africa, relying on PCA results. However, due to common challenges such as missing data and noise, the deployment of dimensionality reduction techniques to analyze ancient DNA or conduct contemporary population genetics studies is controversial. For instance, Yi and Latch [21] demonstrated that missing data can bias PCA analyses. Thus, quantifying the intrinsic uncertainty is pivotal to proper and cautious result interpretation. For example, the relative location of populations/individuals in a PCA/MDS embedding determines its/their relation to other populations or individuals. If, however, there exists a substantial uncertainty with respect to the projection location for some or even all individuals in a study, the respective conclusions may potentially be erroneous and hence misleading. To the best of our knowledge, there currently exists no state-of-the art method to quantify the inherent uncertainty of dimensionality reduction techniques in ancient as well as contemporary population genetic studies. To address this challenge, we introduce Pandora, an open-source tool that estimates the uncertainty of PCA and MDS analyses for population genetics and ancient DNA genotype datasets. Pandora estimates the uncertainty via bootstrapping over the SNPs. In addition to an overall, global stability score, Pandora also estimates the stability of a subsequent k-means clustering based upon the computed embeddings. Both stability scores have values that range between zero and one. Higher values indicate a higher stability and thus a lower uncertainty with respect to the computed embedding for the respective dataset. Beyond this, Pandora also infers dedicated per-individual bootstrap support values for all individuals in the dataset. These support values indicate whether an individual's location in the lower-dimensional embedding space is stable across bootstrapped embeddings, or, if the individual should be considered as being "rogue" when its projections between distinct bootstrapped embeddings differ substantially. For such a "rogue" individual, any conclusion concerning, for instance, the assignment to a specific (sub-)population should be carefully (re-)considered.

In addition to bootstrap-based stability analyses, Pandora can also estimate the stability of dimensionality reduction in the context of whole genome analyses. Via a sliding-window approach, Pandora initially applies dimensionality reduction on subsequent equally sized subsets (windows) along the genome and then compares all pairs of induced per-window embeddings to each other. This allows to quantify the differences among embeddings in different parts of the genome.

Pandora is implemented in Python3 and is available open-source on GitHub `https://github.com/tschuelia/Pandora`. It can be used via the command line or as a Python library. A thorough documentation of the tool, including usage examples, is available at `https://pandorageno.readthedocs.io`.

2 Materials and Methods

Figure 1 depicts our algorithm to estimate the stability of a genotype dataset under dimensionality reduction via bootstrapping. We initially generate N bootstrap replicates by randomly sampling from the set of Single Nucleotide Polymorphisms (SNPs) in the input genotype data *with* replacement. Then, for each bootstrap replicate, we perform a dimensionality reduction using either PCA or MDS, depending on the user's choice. Based on the computed embedding replicates, we subsequently compute the afore-

mentioned three stability values: the overall Pandora stability, the per-individual Pandora Support Values (PSVs), and, *after* applying k-means clustering, the Pandora Cluster Stability (PCS).

An analogous graphical overview for the Pandora sliding-window analyses is provided in Supplementary Figure S1.

In the following, we briefly outline the fundamentals of PCA and MDS, and provide information on the respective implementations we used in Pandora. We also describe Pandora's bootstrap procedure in more detail, focusing on the pairwise comparison of two embeddings, as well as the computation of the PS, PCS, and the PSVs values.

Henceforth, the term 'population' refers to its biological rather than to its statistical interpretation. We refer to a genotype data sequence in the input data as a single individual and, in general, each individual is associated to a population (e.g., "Greek").

2.1 Genotype Data

Pandora was specifically designed for stability analyses of dimensionality reduction on genotype data. This genotype data consists of sequences of SNPs, that is, the genetic variation between individuals. Note that Pandora, following the EIGENSOFT [12] file specification, assumes that data are biallelic. In this case, a specific locus in the maternal and paternal genomic data can be either one of two alleles a or A. Consequently, the genotype of an individual can take one of four states: aa, aA, Aa, or AA. This genetic information is encoded for each SNP and each individual of a genotype dataset under study. Note that the states aA and Aa are typically encoded using the same character, since the resulting genotype is independent of the allele order. Finally, a dedicated special character denotes missing genotypes.

Pandora supports two distinct input data representations: file-based and *NumPy*-based. For the file-based input, Pandora expects three distinct files per dataset: one genotype file containing the sequence data, one file containing metadata for each SNP, as well as one file containing metadata for each individual. Pandora supports the three file formats defined by the EIGENSOFT software package [12] (*EIGENSTRAT*, *ANCESTRYMAP*, *PACKEDANCESTRYMAP*), as well as the *pedigree*, and *binary pedigree* formats defined by the PLINK software package [15]. The *NumPy*-based representation relies on data matrices as defined by the Python library *NumPy* [7]. This allows users to preprocess their dataset using custom (Python) scripts without requiring an explicit (and potentially complicated as well as error-prone) conversion into a specific population genetics file format. Additionally, this input format allows for a more generic deployment of Pandora in other fields of biology beyond population genetics, such as PCA analyses of gene expression data, for instance.

Note that Pandora requires a preprocessed dataset, as it does not internally handle preprocessing. Thus, preprocessing steps, such as pruning Linkage Disequilibrium (LD) or rare variant filtering, must be completed *prior* to conducting Pandora analyses. This prerequisite does not constitute a limitation of Pandora as it applies to any Principal Component Analysis (PCA) or Multidimensional Scaling (MDS) analysis. For instance, EIGENSOFT's *smartpca* tool also expects a preprocessed dataset for performing a PCA analysis. Thus, regardless of the specific analytical tool employed, conducting proper dataset preprocessing constitutes a fundamental step in population genetics research.

2.2 PCA and MDS

PCA and MDS are statistical methods to reduce the dimensionality in data that exhibit a high number of observations/features per individual. The goal is to reduce the dimensionality while preserving as much information as possible, in order to facilitate the interpretability and visualization of high-dimensional input data. While PCA directly operates on the genotype data, MDS reduces the dimensionality of a pairwise distance matrix. This distance matrix can be computed either on a per-individual or per-population basis. A commonly used distance metric is the genetic F_{ST} distance between populations [20] which reflects the proportion of total heterozygosity that can be explained by the within-population heterozygosity.

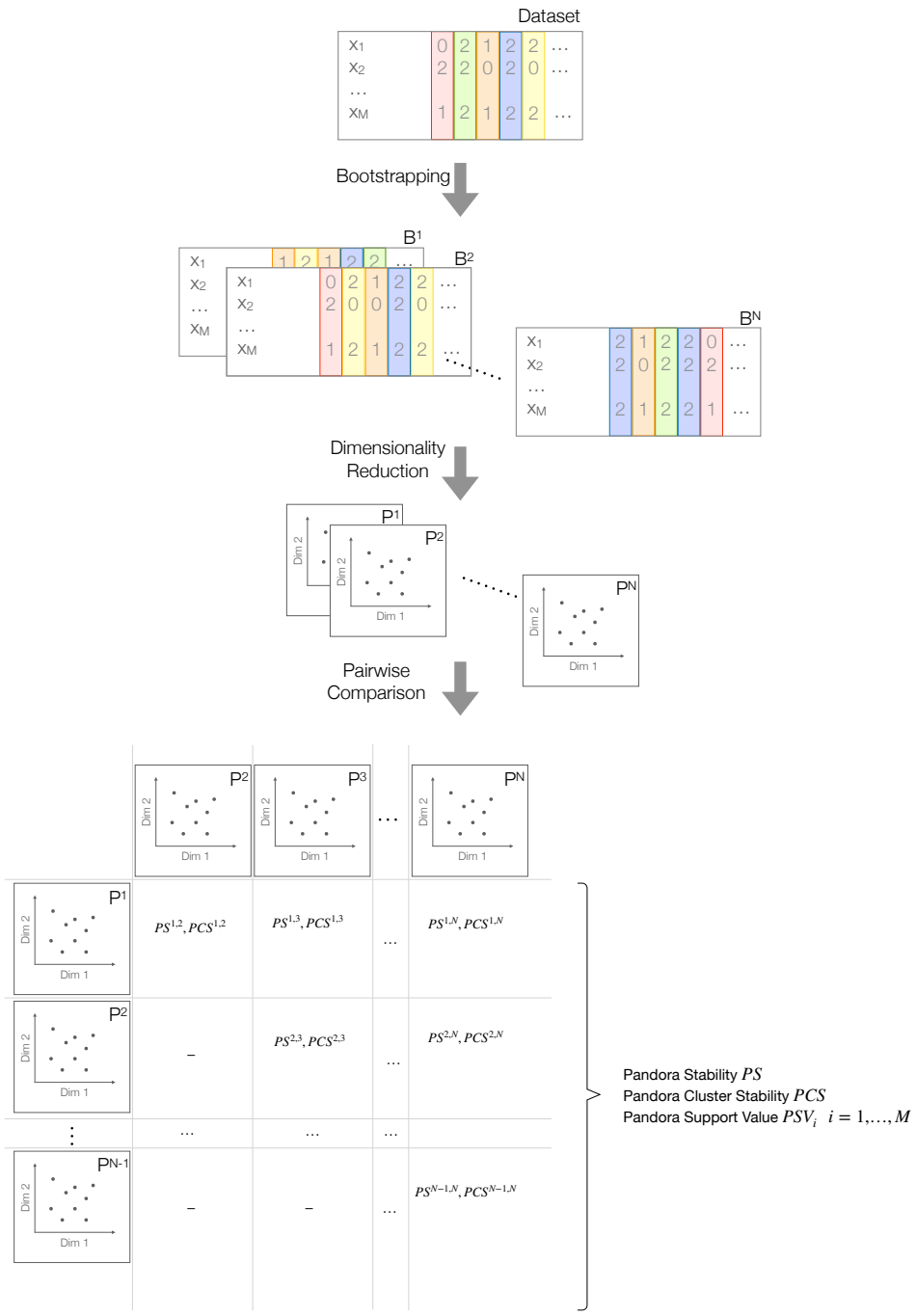


Figure 1: Schematic overview of the bootstrap-based stability analyses for genotype data, as implemented in our Pandora tool.

If the user provides the genotype data using the file-based representation, Pandora computes all PCA embeddings via EIGENSOFT's *smartpca* tool, as it has been specifically designed and highly optimized for population genetics data [12, 14]. Therefore, Pandora supports and propagates all additional *smartpca* PCA analysis commands, including the option to project (ancient) individuals onto an embedding computed using another, distinct set of individuals. For MDS analyses, Pandora computes the F_{ST} distance matrix using *smartpca* and subsequently applies metric MDS as implemented in *scikit-allel* [11].

Using the alternative *NumPy*-based data representation, Pandora performs all PCA analyses using the PCA implementation provided in the Python machine learning library *scikit-learn* [13]. In analogy to the file-based data representation, the default distance metric is the F_{ST} distance (in this case as implemented in *scikit-allel*). Pandora also provides alternative distance metrics such as the Euclidean, Manhattan, or Hamming distances. The subsequent MDS is computed using the *scikit-allel* implementation of metric MDS.

Note that choosing the number of components for PCA or MDS constitutes a challenging and debated question (see Supplementary Material Section 1.2). In analogy to Price et al. [14], we thus set the default number of components for PCA analyses in Pandora to 10. For MDS analyses, we set the default number of components to 2. Pandora users can specify any number of components that best suit their analysis.

To keep the following description of Pandora as generic as possible, we henceforth refer to the result of a PCA or MDS analysis as *embedding*.

2.3 Bootstrapping Genotype Data

The bootstrap procedure [2] is a statistical approach to quantify measures of accuracy based on random sampling with replacement from a set of observations. Estimating the stability of PCA using bootstrapping is not a novel concept. For example, Fisher et al. [5] propose a bootstrap-based method to estimate the sampling variability of PCA results on datasets where the number of per-sample measurements is substantially larger than the than number of samples. However, standard bootstrapping methods are not directly applicable to genotype data. A necessary condition for bootstrapping is the (assumed) independence of the bootstrapped measurements. Individuals (especially individuals of the same species) do, however, generally not evolve independently of each other. However, Felsenstein [4] argues that we may assume independent evolution of genetic loci. Therefore, in Pandora, instead of sampling the individuals, we sample the SNPs s_i to obtain a new bootstrap replicate matrix $B \in \mathbb{R}^{M \times K}$. Thus, a bootstrap replicate has the exact same set of individuals but a distinct SNP composition. In doing so, we follow the standard approach used in phylogenetics, as first proposed by Felsenstein [4].

The computations of embeddings for individual bootstrap replicates are independent of each other, and their computations are therefore straight-forward to parallelize. Pandora exploits this parallelism and allows users to specify the number of threads to use.

Since Pandora bootstraps an embedding (a compute-intensive dimensionality reduction), the computational resource requirements are substantial. To alleviate this, we periodically perform a heuristic convergence assessment until a maximum number of bootstrap replicates is reached. The frequency of this convergence check is determined by the number of threads being used. During the convergence check, Pandora assesses the variation of PS estimates based on random subsets of the computed replicates at the time of the convergence check. When this variation is below a certain tolerance level, Pandora terminates all remaining bootstrap computations and conducts the final stability calculations using the set of completed bootstrap replicates. We provide a detailed description of our convergence criterion in Supplementary Material Section 1.3. The default maximum number of bootstrap replicates in Pandora is 100 and the default convergence tolerance is 0.05. Both settings can be adjusted by the user.

2.4 Stability Estimation

2.4.1 Pandora Stability

The Pandora Stability (PS) describes the overall stability of the genotype dataset that is based on the pairwise similarity scores over all bootstrap replicates. To compute the similarity between two bootstrap embeddings P^u and P^v, we follow the approach suggested by Wang et al. [19] using Procrustes Analysis. Procrustes Analysis determines the optimal transformation $f(P^u)$ that matches the projections of all individuals x_i in P^u to P^v as accurately as possible while preserving the relative distances between individuals. This transformation consists of a scaling factor, a rotation and reflection matrix, and a translation matrix. In Pandora, we use the Procrustes Analysis as implemented in the Python scientific computing package *SciPy* [18]. Prior to computing the optimal transformation, both embeddings are standardized such that $trace(PP^T) = 1$, and both sets of points are centered around the origin. The optimal transformation minimizes the squared Frobenius norm (also called disparity)

$$D^{u,v} = \|f(P^u) - P^v\|_F^2. \tag{1}$$

The disparity $D^{u,v}$ has a minimum of 0 and a maximum of 1 [19]. Based on this *disparity*, we can compute the pairwise *similarity* as

$$PS^{u,v} = \sqrt{1 - D^{u,v}} \tag{2}$$

on a scale from 0 to 1, with higher values indicating higher similarity [19].

To obtain the overall PS, we average over the pairwise similarity across all $\frac{N(N-1)}{2}$ possible distinct pairs of bootstrap replicates

$$PS = \frac{2}{N(N-1)} \sum_{\substack{1 \leq u,v \leq N \\ u<v}} PS^{u,v}. \tag{3}$$

The resulting PS is again a value between 0 and 1. The higher the PS, the more similar the bootstrap replicates are, with 1 indicating that all bootstrap embeddings identically project all individuals.

2.4.2 Pandora Cluster Stability

In analogy to the PS, we initially compute the pairwise Pandora Cluster Stability (PCS) for all unique pairs of bootstrap embeddings and then average over all pairwise scores to obtain the overall PCS

$$PCS = \frac{2}{N(N-1)} \sum_{\substack{1 \leq u,v \leq N \\ u<v}} PCS^{u,v}. \tag{4}$$

To compute the pairwise $PCS^{u,v}$ for bootstrap embeddings P^u and P^v, we first match both embeddings using Procrustes Analysis as described above and subsequently perform an independent k-means clustering for the two embeddings. In order to compare the resulting label assignments for all individuals, we use the same number of clusters k for both embeddings. Finally, we compute the pairwise $PCS^{u,v}$ for bootstrap embeddings P^u and P^v via the Fowlkes-Mallows index [6].

The PCS is a value between 0 and 1. The higher the PCS, the higher is the similarity between clustering results across bootstraps.

Selecting the number of clusters k to best represent the dataset is not trivial. Pandora users can manually set the number of clusters based on, for instance, prior knowledge of the expected underlying (sub-)population structure in the data. When the user does not explicitly specify k, we automatically determine the optimal k based on a grid search and the Bayesian Information Criterion (BIC) [16]. To reduce the computational cost of searching for the optimal k, we determine an input data specific upper bound via the following heuristic. If the individuals of the dataset to be analyzed have population annotations, we set the upper bound to the number of distinct populations in the dataset. If this information is not available, we set the upper bound to the square root of the number of individuals in the dataset. Based on empirical parameter exploration experiments, we set the lower bound to 3. During Pandora development, we observed a substantial influence of k on the resulting PCS scores for various empirical population genetics datasets. We provide further details on these experiments in Supplementary Section 2.5.

2.4.3 Pandora Support Values

To assess the projection stability for each individual in the dataset, we compute the Pandora support for each individual x_i as

$$PSV_i = 1 - \frac{\sum_{\substack{1 \leq u,v \leq N \\ u<v}} \|x_i^u - x_i^v\|}{2N \cdot \sum_{1 \leq u \leq N} \|x_i^u\|}. \quad (5)$$

The fraction in Equation (5) corresponds to the Gini coefficient, a statistical measure of dispersion from economics. In our case, the Gini coefficient measures the dispersion of the projections of an individual with respect to all bootstrap replicates on a scale from 0 to 1. The higher the dispersion, the higher the Gini coefficient will be. To align this with the interpretation of the PS and PCS values, where 0 corresponds to unstable and 1 to stable, we simply subtract the calculated dispersion measure from 1.

2.5 Whole Genome Sliding-Window Analysis

In addition to the above bootstrap stability analyses for genotype data, Pandora also implements a sliding-window stability analysis for whole genome data. The idea is to quantify differences in dimensionality reduction based observations for different subsets/parts of the entire genome. Instead of drawing and comparing bootstrap samples, Pandora splits the input dataset into N overlapping, consecutive subsets (windows) $W^1, \ldots, W^N$. In analogy to the bootstrapping procedure, the default setting is $N := 100$. A more detailed overview of this alternative analysis mode is provided in Supplementary Material Section 1.1.

3 Results

As there exists no ground-truth for empirical datasets to determine the stability estimation accuracy of Pandora, we analyzed 16 simulated genotype datasets published by Yi and Latch [22] with increasing fractions of missing data. We additionally analyzed 19 empirical datasets obtained from various publications with Pandora. Our selection of datasets comprise human genotypes, including ancient DNA data, as well as animal genotypes. For a more detailed overview of the datasets, see Supplementary Material Section 2.1.

In the following, we summarize our key findings for PCA and MDS Pandora analyses on simulated and empirical datasets. We further summarize our analysis of the impact of our bootstrap convergence criterion on the runtime and accuracy of the stability estimates. We provide a detailed analysis setup and result description in Supplementary Material Section 2.

To demonstrate the usage of Pandora in population genetics studies, we provide a more in-depth analysis of one empirical dataset in Supplementary Material Section 2.6.1. The *HO-WE-Çayönü* dataset we use for this exemplary analysis comprises genomic data from modern and ancient individuals. Our analysis demonstrates the particular utility of the PSVs for detecting and preventing potentially erroneous or misleading conclusions. We also present an exemplary sliding-window analysis in Supplementary Material Section 2.6.2.

Note that we provide all scripts to reproduce the dataset assemblies, as well as results, on GitHub `https://github.com/tschuelia/PandoraPaper`. We additionally provide further information on the software versions and hardware we used in Supplementary Material Section 3.

3.1 Stability of Simulated Datasets

To assess if Pandora can detect uncertainty, we estimated the stability of 16 simulated datasets with varying fractions of missing data (0%, 1%, 10%, 20%) [22]. Yi and Latch [22] conclude that PCA is less stable the more data is declared as missing in a dataset. Hence, we expect Pandora to yield lower PS estimates for larger proportions of missing data. As Supplementary Table S3 shows, we do observe this expected behavior.

We further performed MDS analyses, the alternative dimensionality reduction technique available in Pandora, on all simulated datasets. We used two distinct distance metrics: the per-population

distance F_{ST} (Supplementary Table S4) and a per-individual Hamming distance that corrects for missing data (Supplementary Table S5). Our results show a substantial decrease in stability when using the individual-based distance compared to the per-population distance. We thus advise users of Pandora to carefully choose the appropriate distance metric based on the requirements of subsequent downstream analyses and prior knowledge about the underlying data. With the population-based distance metric, all simulated datasets yield a Pandora stability of almost 1, most likely due to the datasets comprising only three distinct populations. Using the per-individual distance metric, we again observe the expected trend of decreasing stability with increasing fractions of missing data (Supplementary Table S5).

Given that Pandora consistently infers lower stability values with higher fractions of missing data under both, PCA and MDS, we conclude that bootstrapping-based stability estimation is appropriate for detecting instability in dimensionality reduction studies.

3.2 Stability of Empirical Datasets

For the PCA analyses on empirical datasets, we used the default *smartpca* settings, that is, 10 principal components and five rounds of outlier detection. For datasets with ancient individuals, we initially computed the PCA using the modern individuals only and subsequently projected the ancient individuals (using the `lsqproject: YES` option in *smartpca*) onto them, prior to conducting the stability estimation. Overall, all 19 empirical datasets are relatively stable under bootstrapping ($PS \geq 0.73$; see Supplementary Table S1). We do, however, observe unstable individuals with low PSVs in *all* datasets. While there is a slight trend for higher PSVs with higher numbers of SNPs, this correlation is not statistically significant (Pearson correlation $r = 0.48$, p-value $= 0.051$). Our results suggest that one should always consider both, the PS, and PSVs when estimating dataset stability as their stability estimates are likely to disagree. In some cases, the PS is substantially higher than the (average) PSVs and vice versa.

In addition to the PCA analyses, we also performed Pandora stability analyses using MDS on all empirical datasets. We used the *smartpca* F_{ST} distance metric between populations as input for the MDS analysis and reduced the data to two dimensions. MDS analyses are more stable than PCA analyses, with a PS of almost 1 across all 19 empirical datasets (see Supplementary Table S2). Yet, analogously to the PCA analyses and despite the overall high PS, we again observe low PSVs for single individuals across all datasets.

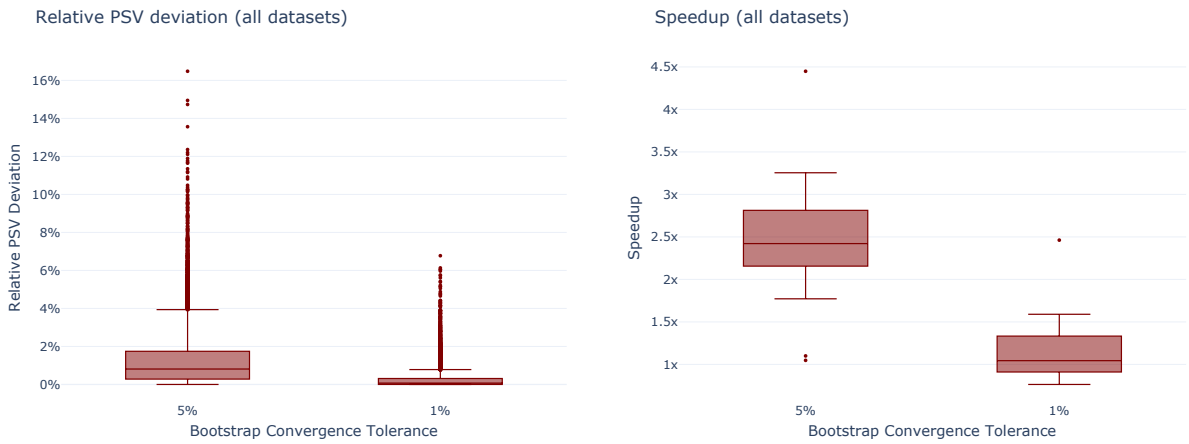
3.3 Speedup and Accuracy under the Bootstrap Convergence Criterion

Using the 19 empirical datasets, we assessed the impact of the implemented bootstrap convergence criterion on the runtime and accuracy of Pandora. We compared the runtime and deviation of PSVs when performing Pandora analysis, by computing the full set of 100 replicates without convergence check and executing Pandora with 5% and 1% convergence tolerance settings.

Aggregated over all datasets, we visualize the distribution of the relative deviation of PSVs (Figure 2a) and the distribution of speedups (Figure 2b) via box plots, at a 5% and 1% convergence tolerance. At 5%, we observe a speedup of 2.46 with almost no loss of accuracy in terms of PS deviation (0% for 12 out of 19 empirical datasets and 1.1% for the remaining 7 datasets). We do however observe PSV deviations of up to 16.5% for a single individual. However, this notable deviation appears to concern only a few samples, as we observe PSV deviations only up to 2.8% when averaged on a per-dataset basis. At 1%, we record an average speedup of only 1.17 with no PS deviation on all datasets and PSV deviations up to 6.8% for a single individual or up to 1.4% when averaged on a per-dataset basis.

Our results suggest that for PCA analyses, the 5% convergence tolerance is suitable when the overall PS is of primary interest. If a correct estimate of the PSVs is paramount, we recommend decreasing the convergence tolerance to 1%.

For MDS analyses, we do not observe a notable speedup despite early-stopping after 20 replicates for



(a) PSV deviations for all individuals of all empirical datasets for both convergence tolerance settings.

(b) Speedup per dataset for both convergence tolerance settings.

Figure 2: PSV deviations and speedups for both analyzed convergence tolerance settings. The box plots show the data for all empirical datasets using PCA analyses.

any dataset under both tolerance settings. Due to the fast computation of MDS compared to PCA (see Supplementary Section 2.4.2), the runtime overhead introduced by the convergence check exceeds the runtime of the relatively fast MDS computation. Consequently, we recommend disabling the bootstrap convergence check for Pandora MDS analyses.

For a more detailed analysis, we refer the interested reader to Supplementary Material Section 2.4.

4 Discussion and Conclusion

Pandora estimates the stability of a genotype dataset under dimensionality reduction using bootstrapping. In addition to an overall stability estimate (PS), Pandora applies k-means clustering to all bootstrapped embeddings to estimate the stability of the assigned clusters across all bootstrap replicates (PCS). Furthermore, Pandora computes a bootstrap support value for each individual in the genotype dataset (PSVs). The latter is particularly useful for datasets with projections of ancient individuals onto an embedding computed on modern individuals only, as the PSVs for the ancient individuals provide a confidence value regarding their placement in the embedding space relative to modern individuals. The lower the support for an individual, the more careful users should be about drawing conclusions regarding this individual based on dimensionality reduction analyses.

Pandora supports PCA and MDS analyses, and provides additional functionality for sliding-window-based stability estimation of dimensionality reduction along whole genome data.

Our analysis results on empirical and simulated data show that Pandora is able to detect instability in PCA and MDS analyses. As expected, Pandora yields lower stability for both, PCA and MDS analyses, with increasing fractions of missing data under simulations. Compared to a single PCA or MDS analysis, Pandora introduces a substantial runtime and resource overhead. Our bootstrap convergence criterion alleviates this to the extent possible.

Our exemplary analysis of the *HO-WE-Çayönü* dataset demonstrates the utility of the Pandora stability analysis, as it helps to prevent potentially erroneous and hence misleading conclusions based on unstable dimensionality reduction. We believe that

Pandora will be of high value to population genetic studies as the inherent uncertainty of PCA and MDS analyses can now be seamlessly and routinely assessed via an easy-to-use tool and thereby contribute to circumvent potentially biased conclusions.

Pandora bootstraps the SNPs and subsequently reduces the dimensionality of the dataset along the same dimension of the dataset, that is, the SNPs. Pandora does not implement a resampling of the individuals (*jackknifing*), as Elhaik [3] demonstrated the sensitivity of PCA studies to (subjective) selection of individuals. The author showed that PCA analyses can be manipulated to confirm the desired hypothesis, depending on the selection of individuals used for PCA analyses. Consequently, for Pandora, we refrained from implementing a jackknifing procedure and thus Pandora results can not directly be used to justify a subjective selection of individuals. However, a jackknifing procedure might allow estimating the bias induced by specific individuals. In conjunction with the Pandora stability estimates, especially the PSVs, this can contribute to identify an objective, meaningful set of individuals for population genetics studies. Devising such a selection criterion will constitute the subject of future work.

As previously mentioned, despite their popularity, the use of PCA or MDS in population genetics is controversial. More modern dimensionality reduction approaches such as t-SNE [17], UMAP [9], or deep learning based variational autoencoders [1] alleviate some problems, but exhibit inherent disadvantages (see Battey et al. [1] for a discussion). Including such methods in Pandora as alternatives to PCA or MDS also remains subject to future work.

5 Acknowledgements

We thank N. Ezgi Altınışık for the very helpful discussion on Pandora, as well as the valuable insights in the methodology of population genetics studies. We thank the aDNA lab at IMBB-FORTH in Heraklion, Crete for the useful initial discussions and ideas. We further thank Jonas Haag for the invaluable help in debugging Python multiprocessing. This work was financially supported by the Klaus Tschira Foundation and by the European Union (EU) under Grant Agreement No 101087081 (Comp-Biodiv-GR).

Funded by the European Union

References

[1] C. J. Battey, G. C. Coffing, and A. D. Kern. Visualizing population structure with variational autoencoders. *G3 Genes—Genomes—Genetics*, 11(1):jkaa036, 2021. doi: 10.1093/g3journal/jkaa036.

[2] B. Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1):1 – 26, 1979. doi: 10.1214/aos/1176344552.

[3] E. Elhaik. Principal component analyses (pca)-based findings in population genetic studies are highly biased and must be reevaluated. *Scientific Reports*, 12, 08 2022. doi: 10.1038/s41598-022-14395-4.

[4] J. Felsenstein. Confidence limits on phylogenies: An approach using the bootstrap. *Evolution*, 39(4):783–791, 1985. doi: 10.1111/j.1558-5646.1985.tb00420.x.

[5] A. Fisher, B. Caffo, B. Schwartz, and V. Zipunnikov. Fast, exact bootstrap principal component analysis for $p > 1$ million. *Journal of the American Statistical Association*, 111(514):846–860, 2016. doi: 10.1080/01621459.2015.1062383.

[6] E. B. Fowlkes and C. L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383): 553–569, 1983.

[7] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, et al. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. doi: 10.1038/s41586-020-2649-2.

[8] J. R. Hughey, P. Paschou, P. Drineas, D. Mastropaolo, D. M. Lotakis, P. A. Navas, M. Michalodimitrakis, J. A. Stamatoyannopoulos, and G. Stamatoyannopoulos. A european population in minoan bronze age crete. *Nature Communications*, 4(1):1861, 2013. doi: 10.1038/ncomms2871.

[9] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[10] P. Menozzi, A. Piazza, and L. Cavalli-Sforza. Synthetic maps of human gene frequencies in europeans. *Science*, 201(4358):786–792, 1978. doi: 10.1126/science.356262.

[11] A. Miles, Pyup.Io Bot, R. Murillo, P. Ralph, J. Kelleher, M. Schelker, R. Pisupati, S. Rae, and T. Millar. GitHub: cggh/scikit-allel v1.3.7, 2023.

[12] N. Patterson, A. L. Price, and D. Reich. Population structure and eigenanalysis. *PLOS Genetics*, 2(12):1–20, 12 2006. doi: 10.1371/journal.pgen.0020190.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[14] A. Price, N. Patterson, R. Plenge, M. Weinblatt, N. Shadick, and D. Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature genetics*, 38: 904–9, 09 2006. doi: 10.1038/ng1847.

[15] S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. de Bakker, M. J. Daly, and P. C. Sham. Plink: A tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics*, 81(3):559–575, 2007. doi: 10.1086/519795.

[16] G. Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461 – 464, 1978. doi: 10.1214/aos/1176344136.

[17] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL `http://jmlr.org/papers/v9/vandermaaten08a.html`.

[18] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[19] C. Wang, Z. A. Szpiech, J. H. Degnan, M. Jakobsson, T. J. Pemberton, J. A. Hardy, A. B. Singleton, and N. A. Rosenberg. Comparing spatial maps of human population-genetic variation using procrustes analysis. *Statistical Applications in Genetics and Molecular Biology*, 9(1), 2010. doi: 10.2202/1544-6115.1493.

[20] S. WRIGHT. The genetical structure of populations. *Annals of Eugenics*, 15(1):323–354, 1949. doi: 10.1111/j.1469-1809.1949.tb02451.x.

[21] X. Yi and E. K. Latch. Nonrandom missing data can bias principal component analysis inference of population genetic structure. *Molecular Ecology Resources*, 22(2):602–611, 2022. doi: 10.1111/1755-0998.13498.

[22] X. Yi and E. K. Latch. Nonrandom missing data can bias principal component analysis inference of population genetic structure. *Molecular Ecology Resources*, 22(2):602–611, 2022. doi: 10.1111/1755-0998.13498.